Netlist and comments for GadgetLab TriLife Blinker circuit layout project.

You usually start with a breadboard, build & test a circuit--for instance a Freeduino connected to some sensors.
When it's working how you intended, you might want to turn the breadboard into a "real" circuit board--for instance a sensor shield that plugs into your Freeduino, or an LCD shield so you're not always redoing the LCD wiring!

EAGLE is a free download (but you can pay to get one that can make larger layouts). You draw traces in EAGLE or a similar program, then can send the files off to get made into a printed circuit board (PCB).

A netlist describes which pins of which parts are connected together--it's used by EAGLE to check that you're drawing the PCB right.
Besides PCB layout, you also see netlists used in circuit simulation software. It's how you describe a circuit to a computer.

LOTS of things are plugged into the 5V or to the Ground nets.
In EAGLE, other nets just get labeled N$1, N$2, N$3 etc. and can connect as few as two things.
A person translating a breadboard would say, each wire plugged into the same row on the breadboard is on the same "net." In EAGLE, they'd open the schematic editor, get the parts out of a library, then wire the parts together according to the netlist they have scribbled on a scrap of paper (or sometimes just reading directly off the breadboard), creating a schematic.

A "schematic" and a "board" are linked together--EAGLE gets mad if you open one without the other (but you can do it) The default setup has the schematic with a white background and the board with a black background.

The reason for the link is that when you have a schematic ready, you can start drawing traces on the board, and EAGLE will know to light up the pad that you're supposed to connect to. Routing the trace around obstacles can be a puzzle, but at least you know where to go. Eagle draws thin yellow 'airwires' going straight from point A to point B, and your goal is to replace every airwire with a nice thick copper trace that doesn't hit any of the wrong nets!

This board will have an 8-pin microcontroller that controls a single LED. It listens to signals coming in on three pins to decide how bright to turn on the LED.

Part lists and functions:

C1 is a capacitor between 5V and ground to filter out noise
U1 is the microcontroller
JP1 is a 6-pin programming connector (the JP stands for "Jumper")
JP2, JP3 and JP4 are 4-pin connectors. I like the excellent Sparkfun footprints that hold onto the connector while you're soldering it.

R1 is a 150-ohm current limiting resistor for the LED
R2, R3, R4 are 200 ohm resistors to protect the microcontroller in case someone plugs in a signal backwards, which could easily happen in this application
LED1 is a bright red LED

Enough background info, here's the actual netlist! Can you wire up the schematic so it matches the netlist? Try on the file GadgetLab_EagleProject_Unwired.brd where I have already picked out all the parts from the library.

| Net | Part | Pad | Pin | Sheet |
|---|---|---|---|---|
| 5V | C1 | 1 | 1 | 1 |
| | JP1 | 2 | 2 | 1 |
| | JP2 | 2 | 2 | 1 |
| | JP3 | 2 | 2 | 1 |
| | JP4 | 2 | 2 | 1 |
| | U1 | 2 | VDD | 1 |
| GND | C1 | 2 | 2 | 1 |
| | JP1 | 3 | 3 | 1 |
| | JP2 | 1 | 1 | 1 |
| | JP3 | 1 | 1 | 1 |
| | JP4 | 1 | 1 | 1 |
| | R1 | P$2 | 2 | 1 |
| | U1 | 7 | VSS | 1 |
| N$1 | JP2 | 4 | 4 | 1 |
| | JP3 | 3 | 3 | 1 |
| | JP4 | 4 | 4 | 1 |
| | LED1 | A | A | 1 |
| | U1 | 3 | GP2/T0CKI/FOSC4 | 1 |
| N$2 | LED1 | K | C | 1 |
| | R1 | P$1 | 1 | 1 |
| N$3 | JP1 | 1 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| | R4 | P$1 | 1 | 1 |
| | U1 | 8 | GP3/~MCLR/VPP | 1 |
| N$4 | JP1 | 4 | 4 | 1 |
| | R3 | P$2 | 2 | 1 |
| | U1 | 5 | GP0/AN0/ICSPDAT | 1 |
| N$5 | JP1 | 5 | 5 | 1 |
| | R2 | P$2 | 2 | 1 |
| | U1 | 4 | GP1/AN1/ICSPCLK | 1 |
| N$6 | JP2 | 3 | 3 | 1 |
| | R4 | P$2 | 2 | 1 |
| N$8 | JP4 | 3 | 3 | 1 |
| | R3 | P$1 | 1 | 1 |
| N$9 | JP3 | 4 | 4 | 1 |
| | R2 | P$1 | 1 | 1 |